UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,123	07/05/2001	Scott Wiltamuth	MSFT-0573/160076.1	5765

41505   7590   01/25/2011
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER
VO, TED T

ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
01/25/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficemonitor@woodcock.com

	Application No.	Applicant(s)
Office Action Summary	09/900,123	WILTAMUTH ET AL.
	Examiner	Art Unit	
	TED T. VO	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>24 September 2010</u>.
2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-3,6,8-13,17,18,23-25,28,30-35,39,40,61-63,66,68-73,77 and 78* is/are pending in the application.
     4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) *1-3,6,8-13,17,18,23-25,28,30-35,39,40,61-63,66,68-73,77 and 78* is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
     a) ☐ All   b) ☐ Some * c) ☐ None of:
       1. ☐ Certified copies of the priority documents have been received.
       2. ☐ Certified copies of the priority documents have been received in Application No. _____.
       3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
     Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
     Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1.   This action is in response to the communication filed on 09/24/2010.

Claims 1-3, 6, 8-13, 17-18, 23-25, 28, 30-35, 39-40, 61-63, 66, 68-73, 77-78 are pending.

Response to Arguments

2.   This is in response to the arguments filed in the remarks on 09/24/2010. The entire of the

specification of the current application directs to a programming specification (spec: p.14-22).

The other pages of the specification is only to mention to the environment of a user who writes a

programming like Fortran, C++, JAVA, C#, etc. Examiner fails to find the patentability in the

specification because it is only to describe a programming specification. This is similarly a guide

programming for how to write correctly a programming syntax. For example, tell the

programmer to recognize syntax difference between lowercase main() and uppercase Main(), etc.

The implementation of a programming segment in a language specification could not fall into

patentability category.

In the present claims, Examiner fails to find any discussion in Applicants' remarks about

the patentability of the claims. The claims are read on a formal act which is used by any

computer user.

Analysis:

implementing <u>in a class</u> (the user uses whatever means, for example, use a text editor in the computer and writes into a class)

a first explicit interface member by explicitly specifying the relationship between the class and the first explicit interface member, the first explicit interface member being excluded from a public interface of the class (an apparatus of a programming specification: In the specification, it admits it is a segment of C#);

<u>implementing in the class a second explicit interface member</u> (the user uses whatever means, for example the text editor in the computer and writes into a second class)

<u>the second explicit interface member having the same signature as the first explicit interface member</u> (an apparatus of a programming specification: In the specification, it admits it is a segment of C#); and

storing said class in a form that includes the implemented first explicit interface member <u>and the implemented second explicit interface member</u> in a computer readable storage medium. (the user selects "save" and selects a directory that resides in a floppy disk, a hard drive to save). Applicants should comply with the requirement that a programming specification should not be patentable. Examiner fails to find a patentable subject matter in the claim, and other claims. It recites only storing a class of a programming specification into a computer readable storage medium.

Claim Rejections - 35 USC § 112

3.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

>   The specification shall conclude with one or more claims particularly pointing out and
>   distinctly claiming the subject matter which the applicant regards as his invention.

4.   Claims 1-3, 6, 8-13, 17-18, 23-25, 28, 30-35, 39-40, 61-63, 66, 68-73, 77-78 are rejected

under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out

and distinctly claim the subject matter which applicant regards as the invention.

         - Claims 1-3, 6, 8-13, 17-18 recite a method. The method appears is to write segments of

a high-level programming segment of a programming language and to store it in a computer

readable medium. However, the claims direct to another scope; it appears an apparatus which is

merely for describing a type of programming language specification rather than an act of the

method:

         *"a first explicit interface member by explicitly specifying the rel*ationship between the

class and the first explicit interface member, the first explicit interface member being excluded

from a public interface of the class" (an apparatus of a programming specification: In the

specification: it admits it is a segment of C#);

         "the second explicit interface member having the same signature as the first explicit

interface member *"* (an apparatus of a programming specification: In the specification: it admits

it is a segment of C#).

The claims are indefinite because it mixes the claim types. The limitation, a first explicit interface member by explicitly specifying the relationship between the class and the first explicit interface member" and

"the second explicit interface member having the same signature as the first explicit interface member",

are only programming declarations of the C#, admitted by Applicants. With this type of claiming, the two claim languages above do nothing with the act <u>implementing in the class</u>, but they aim to associate with the instructions of c# specification.

(It should be note that programming specification such as C, Java, FORTRAN are only programming; the programming language itself is not an application. So is C#).

The functionality of the claimed phases above in light of the specification is only a programming segment complied with syntax requirement of a proper programming language. Thus, it is unable to identity whether the claim is claiming a method for storing classes of a programming specification or the apparatus of the programming specification. Such of claiming is indefinite because it does not clear applicants want to claim a protection of a method of a programming specification. Note: See IPXL Holdings v. Amazon.com, Inc., 430 F.2d 1377, 1384, 77 USPQ2d 1140, 1145 (Fed. Cir. 2005); Ex parte Lyell, 17 USPQ2d 1548 (Bd. Pat. App. & Inter. 1990) (claim directed to an automatic transmission workstand and the method of using it held ambiguous and properly rejected under 35 U.S.C. 112, second paragraph)).

All the dependent claims 2-3, 6-15, 17-19 appear direct to the C# specification rather a novelty method.

E.g. Claim 2: A method according to claim 1, wherein said specifying of the relationship includes specifying a qualified name of the class.

The claim attempts to use the term specifying, that is mentally only in user's mind, but recursive with a syntax requirement of the programming language specification: a qualified name of the class.

Further e.g. claim 3 *"includes specifying an interface name and said at least one interface member name"*, where

includes specifying merely read the hand writing, keyboard typing in to a text editor, and

an interface name and said at least one interface member name, merely a syntax specification of the C#. Thus it is unable to determine whether applicant to claim **specifying** or a programming language specification.

Claims 6-15, 17-19 are indefinite because the claims are apparatus of programming specification.

- Claims 23-25, 28, 30-35, 39-40 recite a scope of a computer readable storage medium to store executable instructions.

However, the claimed subject, "instructions" is indefinite because it does not point out this particularly "instruction" and what it does. The claims recite away from functionality of claimed readable medium, but merely associate with a high level programming specification. The scope of the claim is unclear, whether the claims direct to a medium storing instruction or a high level programming specification. Examiner interprets it is a computer readable storage medium storing instructions.

- Claims 61-63, 66, 68-73, 77-78 recite a method generating an object using a compiler. However, the fact is, implementing the members of the class remains a high level class that can use pen or text in an editor to implement. It should be noted that the function of compiler is to generate executable code from high level specification via several compiling phases; however, the claimed recitation is only implement a class in which a compiler is no needed. The Claims attempt to use compiler but it does not show compiling. Its claiming in term of enablement is in question. The functionality in the claims is ambiguous, and the claimed subject matters in the claims are unclear.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A person shall be entitled to a patent unless –

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6.   Claims 1-3, 6, 8-13, 17-18, 23-25, 28, 30-35, 39-40, 61-63, 66, 68-73, 77-78 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joseph Bergin, "Multiple Inheritance in

Java", June 2000, http://csis.pace.edu/~bergin/patterns/multipleinheritance.html and

http://web.archive.org/web/20000816190515/http://www.csis.pace.edu/~bergin/patterns/multiple

inheritance.html, 4 pages, in view of Alex Shindich, "A few words about Python interfaces", an

article posted in Python-list, http://code.activestate.com/lists/python-list/212738/, on 4-15-2001,

4 pages and Prescod, "Python Interface Declaration Language", copyright 1998, W3C, retrieved

from http://www.prescod.net/pytypes , 13 pages.

As per claim 1: Bergin discloses (Currently amended)

A method comprising:

> implementing in a class (See p. 1 or p. 2: MProvides, or see p. 4: class OtherInterface) a first
> explicit interface member by explicitly specifying the relationship between the class and the
> first explicit interface member, the first explicit interface member being excluded from a
> public interface of the class;

> See p. 4: class member: ParentChild, that specifies the relationship between "child", each is
> an interface member within the class parent. The term Private "Other interface child"
> declared in p. 4 has the means the "Other child" has Mixin inherence from ParentChild, but it
> is publicly excluded from other interfaces of the "child".

> implementing in the class a second explicit interface member, the second explicit interface
> member having the same signature as the first explicit interface member; and

> See p. 4, the member "parent child" . Either Parent child or Other Child has the same
> signature child member to its parent.

> storing said class in a form that includes the implemented first explicit interface member and
> the implemented second explicit interface member in a computer readable storage medium.

> Bergin shows the class MProvides or Class OtherInterface is created in a form of a program

(Class) that is implemented with interface members.

Bergin proposes a Mixin inheritance of a class in an interface, where the class is used with declaration "interface": For example: interface MRequires Or Interface OtherInterface.

Bergin's proposal does not say this type of implementation is "explicit interface" and explicit interface member.

Shindich discloses that in Python, the type of interface which has a certain type hierarchal structure could be defined within inheritance, but required an explicit implement. That is "explicit interface" and declaration of interface members within the class is explicit interface member implementation (See Shindich, class Father, mother, member: Son1, Son2, or S1, S2. and the inheritance is Son, they have same signature, but some of the member implementation might excluded from other. E.g. s1.Playviolin() and s1.thinkHard(). However, s2.Playviloin() (sic) and s2.hackCode()) . Shindich admits the code is Python.

Thus, it would be obvious to an ordinary in the art at the time to include the Shindich to provide a term of programming formation with the term "explicit interface". The same formation is used in the proposal of Bergin for include Mixin inheritance in interface implementation of class. The combination is necessary for suggest a same formation but term mentioning.

On the other hand, Prescod discloses Python programming which using declaration for "Interface". Python allows programmer to declare a specific interface (See p. 8) which is clearly incorporated with the freedom of Python syntax for use in the Mixin inheritance of Shindich. More importantly, Prescod mentions every form of Python program will require storing in a computer and being compiled by a compiler.

Thus, it is obvious the ordinary in the art at the time to further include the teaching Python

and its interface implementation and its using a compiler to compile the stored program in a

memory for addressing that the use of Explicit Interface in Python of Shindich is incorporated

with, and storing a class in a form in a computer readable medium is conforming to the

requirement.

(Claims 2-3, 6, 8-13, to 17-18 are directing to programming entity, the claims are mixed with

apparatus, i.e. programming specification. So are claims 23-25, 28, 30-35, 39-40, 61-63, 66, 68-

73, 77-78).

As per claim 2: Regarding limitation,

The method according to claim 1, wherein said specifying of the relationship between the class

and the first explicit interface member includes: specifying a qualified name of the class.

The term "specifying" causes it looks like an action. However, the use the claim phase merely

reads on a program pro se. (See Bergin The Mixin: p. 2-3: the term used within public and

private, or see p. 4. Other child).

As per claim 3: Regarding limitation, **The method according to claim 2, wherein said specifying**

of the qualified name includes: specifying an interface name and a name of the first explicit

interface member. The use the claim phase merely reads on a program pro se. (See Bergin The

Mixin: p. 2-3: the term used within public and private, or see p. 4. the class OtherInterface).

As per claim 6: Regarding limitation, The method according to claim 1, wherein implementing in a class the first explicit interface member comprise: implementing in the class an internal interface not accessible to a consumer of said class.

The use the claim phase merely reads on a program pro se. (See Bergin The Mixin: p. 2-3: the term used within public and private, or see p. 4. the class OtherInterface, Child and Other Child).

As per claim 8: Regarding limitation, The method according to claim 1, wherein the second explicit interface member has the same return type as the first explicit interface member.

The use the claim phase merely reads on a program pro se. (See Bergin The Mixin: p. 2-3: the term used within public and private, or see p. 4. the class OtherInterface, Child and Other Child).

As per claim 9: Regarding limitation, The method according to claim 1, wherein the second explicit interface members is included in a public interface of the class.

(See Bergin The Mixin: p. 2-3: the term used within public and private, or see p. 4. the class OtherInterface).

As per claim 10: Regarding limitation, A method according to claim 1, wherein the first explicit interface member comprises a first version of a generic interface, and the second explicit interface member comprises a second version of the generic interface.

The use the claim phase merely reads on a program pro se. (See Bergin p. 4. Within class OtherInterface, Class OtherChild and Class ParentChild. Or see in Shindich, Class s1 and class s2).

As per claim 11: Regarding limitation, The method according to claim 1, wherein the class is programmed according to an object-oriented programming language.

The use the claim phase merely reads on a program pro se. All prior arts are related to OO languages.

As per claim 12: Regarding limitation, The method according to claim 1, wherein an implementation of an explicit interface member is a method, property, event, or indexer declaration that references a fully qualified interface member name.

The use the claim phase merely reads on a program pro se. (See Bergin p. 4. Within class OtherInterface, it has methods. Or see in Shindich, Class s1.thinkhard()).

As per claim 13 Regarding limitation,

The method according to claim 1, wherein the class names an interface in a base class list of the class that contains a member whose fully qualified name, type, and parameter types exactly match those of the implementation of the first explicit interface member.

The use the claim phase merely reads on a program pro se. (See Bergin The Mixin: p. 2-3: the term used within public and private, or see p. 4. the class OtherInterface, Child and Other Child).

As per claim 17: Regarding limitation, The method according to claim 1, wherein it is not possible to override the first explicit interface member wherein the first explicit interface member calls another virtual method, and wherein a class derived from the class overrides the first explicit interface member.

The use the claim phase merely reads on a program pro se, where Bergin proposes the Mixin

interface in JAVA. Thus the language of JAVA or PYTHON allows programmer to declare

specific syntax to perform the specific programming code of the claim.

As per claim 18: Regarding limitation, The method according to claim 1, wherein the class

re-implements an interface of the first explicit interface member by including the interface

member in the base class list of the class.

The use the claim phase merely reads on a program pro se. (See Bergin The Mixin: p. 2-3: the

term used within public and private, or see p. 4. the class OtherInterface, Child and Other Child).

As per claims 23-25, 28, 30-35, 39-40: Claim is read on a generic computer readable medium,

such as a hard drive. However, for the use of language specification recited in the claims, see

related rationale addressed in claims 1-3, 6, 8-13, 17-18.

As per claims 61-63, 66, 68-73, 77-78: See related rationale addressed in claims 1-3, 6, 8-13, 17-

18

Conclusion

7.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The

examiner can normally be reached on 8:00AM to 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is

assigned is the Central Facsimile number 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be

directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of

an application may be obtained from the Patent Application Information Retrieval (PAIR)

system. Status information for published applications may be obtained from either Private PAIR

or Public PAIR. Status information for unpublished applications is available through Private

PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business

Center (EBC) at 866-217-9197 (toll-free).

TTV
January 07, 2011

/Ted T. Vo/
Primary Examiner, Art Unit 2191